

Quantum Phase Estimation Algorithm for Finding Polynomial Roots

T. Tansuwannont,^{1,2} S. Limkumnerd,^{1,3} S. Suwanna,² and P. Kalasuwan^{4,*}

¹*Department of Physics, Faculty of Science, Chulalongkorn University
254 Phayathai Road, Pathumwan, Bangkok, 10330, Thailand*

²*Collaborative Research Unit on Quantum Information & Department of Physics,
Faculty of Science, Mahidol University, Rama VI Road, Ratchathewi, Bangkok, 10400, Thailand*

³*Research Center in Thin Film Physics, Thailand Center of Excellence in Physics,
CHE, 328 Si Ayutthaya Rd., Bangkok 10400, Thailand*

⁴*Department of Physics, Faculty of Science, Prince of Songkla University, Hat-Yai, Songkhla, 90112, Thailand*

(Dated: October 16, 2015)

Quantum algorithm is an algorithm for solving mathematical problems using quantum systems encoded as information, which is found to outperform classical algorithms in some specific cases. The objective of this study is to develop a quantum algorithm for finding the roots of n th degree polynomials where n is any positive integer. In classical algorithm, the resources required for solving this problem increase drastically when n increases and it would be impossible to practically solve the problem when n is large. It was found that any polynomial can be rearranged into a corresponding companion matrix, whose eigenvalues are roots of the polynomial. This leads to a possibility to perform a quantum algorithm where the number of computational resources increase as a polynomial of n . In this study, we construct a quantum circuit representing the companion matrix and use eigenvalue estimation technique to find roots of polynomial.

I. INTRODUCTION

Roots finding is a centuries-old problem that has continued to attract considerable research interests and efforts due to its relevance in many fields of mathematics and physics involving geometry, number theory, probability and combinatorics. It is well known that for a polynomial of degree 4 or less, there exists a formula or procedure to solve for its roots exactly¹. However, such a task is impossible for a polynomial of degree 5 or greater². Many root-finding algorithms have been devised for obtain approximated roots of a polynomial of arbitrary degree^{3–5}.

The plausibility of a quantum computer—a new type of computation, which embraces quantum mechanics into its information, algorithms, and output measurements—has entailed quantum algorithms. A simple enhancement by rather non-intuitive mathematics of quantum mechanics like superposition, the uncertainty principle, and entanglement, brings quantum algorithms forth to a new level of computation unreachable before by conventional computers with classical algorithms. For example, the Shor's algorithm, an algorithm to factorize a large integer into a product of primes, is proven in principles to overcome the classical-algorithm limit in terms of speed⁶. A breakthrough in quantum simulation is expected to bring eminent impact into science and technology^{7,8}. Recent progress on the actual quantum devices, such as a successful small-molecule simulation^{9–12} or probing the statistics of quantum systems^{13–15}, have yielded high promises and attracted immense interests. The advancement of computation and simulation in the aforementioned examples owes largely to a common underlying method called *phase estimation algorithm* (PEA)¹⁶. Still, PEA can be improved, especially at the fundamental algorithm of finding roots of a polynomial.

However, PEA is only applicable to unitary operators which are not always the case for some quantum algorithms; for instance, the phase measurement under the circumstance where decoherence is present in the process¹⁷. In a measurement process of the quantum algorithm, the non-unitary matrices also play key roles as projective operators. In order to modify existing PEA to be suitable for eigenvalue problems comprehensively, a programmable circuit, and measurement of the control and ancillary qubits are recently exploited to tailor-made any arbitrary matrix¹⁸. The great advantage of this proposed scheme is that any matrix can be constructed and the control gate of the respective matrix can be realized, paving ways to build a quantum computer which can calculate eigenvalue of any matrix. However, some drawbacks exist as the algorithm itself may not be efficient for complicated matrices, which quantum complexity arises following the increasing number of non-zero matrix elements¹⁹. Further investigation on the algorithm in terms of appropriate complexity is still needed.

Our main aim in this paper is to propose a modified quantum phase estimation algorithm for finding polynomial roots, where we present a benchmark implementation of quantum non-unitary eigenvalue calculation scheme for polynomials. This specific task represents the least complex eigenvalue, which the algorithm can be fruitful without too much concern over the complexity.

This article is organized as follows. The remaining subsections of this section will cover key concepts and ideas about the phase estimate algorithm, and the iterated phase estimate algorithm (IPEA) for unitary operators, as well as the quantum algorithm to find complex eigenvalues of a general matrix. In Section II, we present our modified PEA and IPEA, together with the companion matrix approach, and more importantly, the circuit design to estimate roots of a polynomial of degree

n . There we focus our presentation of the circuit operation and outputs, leaving the discussion and complexity analysis in Section III. Finally, the conclusions are summarized in Section IV.

A. PEA and IPEA for Eigenvalue Problems of Unitary Operators

In the original version of PEA, a phase φ arising after a unitary evolution U with eigenvalue $\exp(2\pi i\varphi)$ is operated on its basis. Because a quantum evolution can be interpreted by a phase factor $U = \exp(-i\mathbf{H}t/\hbar)$, where \mathbf{H} is a Hamiltonian of a finite system, the phase as a result of the phase estimation algorithm is indeed the eigenvalue of the Hamiltonian. PEA has also been introduced as a potential quantum tool to effectively solve various eigenvalue problems involving unitary operators^{20–23}. The unitary operators play a central role in all of the quantum algorithms, as they are required for universal quantum computations²⁴.

In order to estimate the value of a phase parameter ω_j up to the b bit-precision using PEA, b ancillary qubits in control register are required. In practice, however, the number of qubits which can be implemented is very limited. Iterative Phase Estimation Algorithm (IPEA) is an algorithm improved from the original PEA with an aim to estimate ω_j up to b^{th} digit while using only one ancillary qubit together with b iterations as a result of scalable inverse quantum Fourier transform in a semi-classical manner^{25,26}. In order to explain the algorithm as illustrated in Fig. 1, we first assume that the phase parameter ω_j has a binary expansion no more than b digits (written as $\omega_j = 0.x_1x_2x_3\dots x_b000\dots$). Initially, all of the ancillary qubits are prepared in state $|0\rangle$ and the target register is prepared in the eigenstate $|\psi_j\rangle$ of unitary operator U . A Hadamard gate is applied to the control register in order to prepare state $\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)$. In the first iteration ($k = 1$), a $c-U^{2^{b-1}}$ and

$$Z(\theta_k) = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\theta_k} \end{bmatrix}, \quad (1)$$

where $\theta_1 = 0$ are applied. After that, the second Hadamard gate is applied on the control qubit and its state is measured in the computational basis $\{|0\rangle, |1\rangle\}$. This results in state

$$\frac{1}{2}[(1 + e^{2\pi i(0.x_b)})|0\rangle + (1 - e^{2\pi i(0.x_b)})|1\rangle], \quad (2)$$

whose measurement gives either 0 or 1, and is determined by the majority probability between $|0\rangle$ and $|1\rangle$. This measurement result consequently dictates the value of x_b . The next iteration is performed with the $c-U^{2^{b-k}}$ and $Z(\theta_k)$, where $\theta_k = 2\pi(0.0x_{b-k+2}x_{b-k+3}\dots x_b)$ is calculated by the feed-forwarded measurement result of the prior iterations up to x_{b-k+1} . The algorithm is finished when the digit x_1 is obtained.

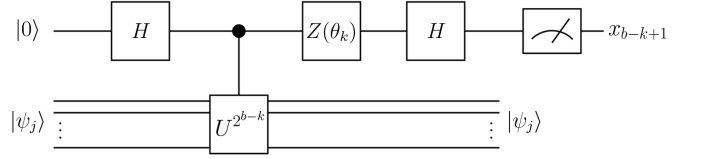


FIG. 1. A circuit for the k^{th} iteration of the IPEA where θ_k is the feedback from prior iterations.

The original IPEA has been used to determine only the phase parameter ω_j of the eigenvalue $\lambda_j = e^{2\pi i\omega_j}$ of a unitary matrix U . In general, however, an eigenvalue of a non-unitary operator can be written as $\lambda_j = |\lambda_j|e^{2\pi i\omega_j}$. The standard IPEA therefore cannot be applied without the knowledge of modulus $|\lambda_j|$.

B. Quantum Algorithm for Finding Complex Eigenvalues of General Matrices

Recently, Daskin *et al.* have introduced their technique to find the complex eigenvalues of general matrices¹⁸. In order to employ the IPEA on the non-unitary operators, first of all, the non-unitary operator O has to be controlled by a phase qubit and the $c-U^{2^{b-k}}$ in Fig. 1 is replaced by $c-O^{2^{b-k}}$.

In the scheme proposed by Daskin *et al.*, the decomposition of the control gate of the non-unitary operator O of size $N = 2^m$ uses the programmable circuit design, which requires $m + 1$ ancillary qubits and m main qubits¹⁸. The operator O generally has a eigenvalue of the form $|\lambda_j|e^{2\pi i\omega_j}$. In case that $\omega_j = 0.x_1000\dots$, an operation of $c-O$ followed by the Hadamard gate gives an output state

$$\left[\left((1 + |\lambda_j|e^{2\pi i(0.x_1)})|0\rangle_p + (1 - |\lambda_j|e^{2\pi i(0.x_1)})|1\rangle_p \right) |\psi_j\rangle_m \right], \quad (3)$$

where p and m denote phase qubit and main qubits, respectively. As can be seen, the $c-O^{2^{b-k}}$ can be realized by the decomposition proposed by Daskin *et al.* It is also possible to estimate its phase ω_j via the IPEA from the probability shown in (3) in the same fashion as the phase estimation results determined by (2). However, the true novelty of the scheme is in the estimation of $|\lambda_j|$ —taking the calculation to a complete eigenvalue estimation for any non-unitary matrix. Following the result shown in (3), the value of $|\lambda_j|$ is related to the probability P_0 or P_1 of finding the phase qubit in states $|0\rangle$ or $|1\rangle$, respectively. Let $P = \max\{P_0, P_1\}$, so that we can estimate $|\lambda_j|$ as

$$|\lambda_j| = 2N^2\sqrt{P} - 1, \quad (4)$$

where N is the dimension of matrix. In practice, $|\lambda_j|$ is determined by the statistics of the measurement. We can also improve the accuracy of the estimation by using

the statistics from other iterations. For the k^{th} iteration after which $c\text{-}O^{2^{b-k}}$ is operated followed by $Z(\theta_k)$ and the Hadamard gate, the relationship between $P^{(k)}$ and $|\lambda_j|$ becomes

$$|\lambda_j|^{2^{b-k}} = 2N^2 \sqrt{P^{(k)}} - 1. \quad (5)$$

Since we can estimate both $|\lambda_j|$ and ω_j , the complex eigenvalue λ_j can be determined.

II. QUANTUM ALGORITHM FOR FINDING POLYNOMIAL ROOTS

A. Companion Matrix Approach

As the aim of this study is to find roots of a generic polynomial of degree n , we can formulate this problem as the eigenvalue problem of a non-unitary operator. First of all, consider

$$p(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0; \quad (6)$$

it can be factorized into the form

$$p(x) = (x - z_1)(x - z_2) \dots (x - z_n), \quad (7)$$

where $z_1, z_2, \dots, z_n \in \mathbb{C}$ are the roots of $p(x)$. From general linear algebra²⁷, the roots of polynomial $p(x)$ are eigenvalues of its companion matrix defined as

$$C_p = \begin{bmatrix} 0 & 1 & 0 & & \\ & 0 & 1 & 0 & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ -a_0 & -a_1 & \dots & -a_{n-2} & -a_{n-1} \end{bmatrix} \quad (8)$$

with respect to the basis $\{1, x, x^2, \dots, x^{n-1}\}$.

Daskin's algorithm requires that the absolute value of every coefficient a_i must be less than or equal to 1, since rotation gates are used to simulate these coefficients. Therefore, we introduce a scaling method to meet this requirement. Let a_{\max} denote the greatest absolute value of a_0, a_1, \dots, a_n . We choose a basis of circuit in the x -mode or $(1/x)$ -mode depending on whether $|a_n|$ or $|a_0|$ is greater to maximize the success probability of the circuit scheme.

In case $|a_n| > |a_0|$, the x -mode will be chosen, so the polynomial $p(x)$ can be equivalently expressed in the form

$$\frac{a_n}{a_{\max}}x^n + \frac{a_{n-1}}{a_{\max}}x^{n-1} + \dots + \frac{a_1}{a_{\max}}x + \frac{a_0}{a_{\max}} = 0. \quad (9)$$

Let $\mu = \frac{a_{\max}}{a_n}$ be a scaling factor. Then the corresponding eigenvalue equation is written as

$$\begin{bmatrix} 0 & \frac{a_n}{a_{\max}} & 0 & & \\ & 0 & \frac{a_n}{a_{\max}} & 0 & \\ & & \ddots & \ddots & \\ & & & 0 & \frac{a_n}{a_{\max}} \\ -\frac{a_0}{a_{\max}} & -\frac{a_1}{a_{\max}} & \dots & -\frac{a_{n-2}}{a_{\max}} & -\frac{a_{n-1}}{a_{\max}} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ \vdots \\ x^{n-2} \\ x^{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \frac{1}{\mu} & 0 & & \\ & 0 & \frac{1}{\mu} & 0 & \\ & & \ddots & \ddots & \\ & & & 0 & \frac{1}{\mu} \\ -a'_0 & -a'_1 & \dots & -a'_{n-2} & -a'_{n-1} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ \vdots \\ x^{n-2} \\ x^{n-1} \end{bmatrix} \\ = \frac{x}{\mu} \begin{bmatrix} 1 \\ x \\ \vdots \\ x^{n-2} \\ x^{n-1} \end{bmatrix}. \quad (10)$$

where $a'_i = \frac{a_i}{a_{\max}}$. The eigenvalue of modified companion matrix is x/μ .

On the other hand, if $|a_0| > |a_n|$, the $(1/x)$ -mode will be used. Dividing the polynomial $p(x)$ by $a_{\max}x^n$ leads to

$$\frac{a_0}{a_{\max}}\left(\frac{1}{x}\right)^n + \frac{a_1}{a_{\max}}\left(\frac{1}{x}\right)^{n-1} + \dots + \frac{a_{n-1}}{a_{\max}}\left(\frac{1}{x}\right) + \frac{a_n}{a_{\max}} = 0. \quad (11)$$

In this case, a scaling factor is $\mu = \frac{a_{\max}}{a_0}$, and the corresponding eigenvalue equation is in the form

$$\begin{bmatrix} 0 & \frac{a_0}{a_{\max}} & 0 & & \\ & 0 & \frac{a_0}{a_{\max}} & 0 & \\ & & \ddots & \ddots & \\ & & & 0 & \frac{a_0}{a_{\max}} \\ -\frac{a_n}{a_{\max}} & -\frac{a_{n-1}}{a_{\max}} & \dots & -\frac{a_2}{a_{\max}} & -\frac{a_1}{a_{\max}} \end{bmatrix} \begin{bmatrix} 1 \\ 1/x \\ \vdots \\ 1/x^{n-2} \\ 1/x^{n-1} \end{bmatrix} \\ = \begin{bmatrix} 0 & \frac{1}{\mu} & 0 & & 1 \\ & 0 & \frac{1}{\mu} & 0 & \\ & & \ddots & \ddots & \\ & & & 0 & \frac{1}{\mu} \\ -a'_0 & -a'_1 & \dots & -a'_{n-2} & -a'_{n-1} \end{bmatrix} \begin{bmatrix} 1 \\ 1/x \\ \vdots \\ 1/x^{n-2} \\ 1/x^{n-1} \end{bmatrix} \\ = \frac{1}{\mu x} \begin{bmatrix} 1 \\ 1/x \\ \vdots \\ 1/x^{n-2} \\ 1/x^{n-1} \end{bmatrix}, \quad (12)$$

where $a'_i = \frac{a_{n-i}}{a_{\max}}$ in this case. Note that the eigenvalue of the modified companion matrix is $1/\mu x$.

However, the traditional companion matrix as described in (8) has 1's in the upper diagonal entries but all of such entries of the modified companion matrices as shown in (10) and (12) have absolute values less than 1. To rectify this, we introduce a scaling gate $S_{m,\mu}$ which will be explained in details later; see Equation (21).

B. Quantum Circuit Design

Our design of the respective algorithm relies on Polynomial Representative Circuit (PRC), a circuit to represent this modified companion matrix as illustrated in Fig. 2. PRC requires m main qubits and 2 ancillary qubits where $2^m = n$ is a degree of the polynomial. (Although it is inconvenient, the circuit is also applicable for $n \neq 2^m$ simply by shifting the degree of polynomial up to the nearest power of 2.) First, let the main qubit be prepared in the initial state :

$$|\alpha\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-2} \\ \alpha_{n-1} \end{bmatrix}, \quad (13)$$

and we define $|\beta\rangle$ as a result of C_p operating on $|\alpha\rangle$; i.e.

$$C_p|\alpha\rangle = |\beta\rangle. \quad (14)$$

Multiplying $|\alpha\rangle$ by the modified companion matrix from (10) or (12) gives $|\beta\rangle$ in the form:

$$|\beta\rangle = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n-1} \\ -\vec{a}' \cdot \vec{\alpha} \end{bmatrix}, \quad (15)$$

where $\vec{a}' \cdot \vec{\alpha} = a'_0\alpha_0 + a'_1\alpha_1 + \dots + a'_{n-1}\alpha_{n-1}$. Similar to the circuit introduced by Daskin *et al.*, our circuit consists of *Input Modification Block*, *Formation Block*, and *Combination Block*. The main ingredient of the Input Modification Block is a cyclic-swap gate C_s applied on the main qubits. The matrix representation of the gate is

$$C_s = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & \ddots & \ddots & \\ 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}, \quad (16)$$

which can be implemented by the Toffoli gates as shown in Fig. 3.

The aim of the operator C_s is to generate the matrix element of the companion matrix from row 1 to row $n - 1$. Since the operation is underpinned by the presences of the sequences of Toffoli gates, the algorithm will be plagued by the huge complexity. the complexity of the algorithm is quite large, and yet still smaller than that of Daskin's scheme, as their matrix elements are generated by the formation block which incurs more complexity.

The formation block in our version plays a role of the controlled gate of an operator F_μ , which represents the components in the last row of the modified companion

matrix C_p as in (10) or (12). The rotation gate R_i is represented by a matrix as follows:

$$R_i = \begin{bmatrix} a'_i & \sqrt{1 - a'^2_i} \\ -\sqrt{1 - a'^2_i} & a'_i \end{bmatrix}, \quad (17)$$

where $i = 0, 1, 2, \dots, n - 1$. Accordingly, the array of R_i forms the block matrix F_μ as follows:

$$F_\mu = \begin{bmatrix} R_1 & & & \\ & \ddots & & \\ & & R_{n-1} & \\ & & & R_0 \end{bmatrix}. \quad (18)$$

The operation of F_μ can be simulated by a sequence of controlled-rotation gates as in Fig. 4. The operation of F_μ will be performed on main qubits and the second ancillary qubit in case that the state of first ancillary qubit is $|1\rangle$.

Next step, in the combination block, we define the operator C as follows:

$$C = (XH)^{\otimes m} \otimes I = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \dots & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \dots & \bullet & \bullet \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \bullet & \bullet & \bullet & \bullet & \dots & \bullet & \bullet \\ 1 & 0 & 1 & 0 & \dots & 1 & 0 \\ \bullet & \bullet & \bullet & \bullet & \dots & \bullet & \bullet \end{bmatrix}. \quad (19)$$

where " \bullet " represents the elements we can neglected because they will be filter out by post-selection at the final stage of the algorithm.

There are three sub-tasks to undertake in the combination block. First, a controlled- C gate with the operator C is operated on the main qubits conditioning to the state of the first ancillary qubit as $|1\rangle$ to create the $\vec{a}' \cdot \vec{\alpha}$ component. As a result, the operation of C_s , F_μ , and C on main qubits and second ancillary qubit conditioning to the first ancillary state $|1\rangle$ give the following output:

$$\frac{1}{\sqrt{2^m}} \begin{bmatrix} \bullet \\ \bullet \\ \vdots \\ \bullet \\ \vec{a}' \cdot \vec{\alpha} \\ \bullet \end{bmatrix} \quad (20)$$

with the probability amplitude $\frac{1}{\sqrt{2^m}}$. This amplitude is required to be balanced with the case where the state of the first ancillary qubit is $|0\rangle$. At this stage of the algorithm, the scaling gate $S_{m,\mu}$ is introduced to balance this probability and generate $1/\mu$ in the upper diagonal entries of the modified companion matrix. It is defined as

$$S_{m,\mu} = \frac{1}{\sqrt{2^m}\mu} \begin{bmatrix} 1 & \sqrt{2^m\mu^2 - 1} \\ -\sqrt{2^m\mu^2 - 1} & 1 \end{bmatrix}. \quad (21)$$

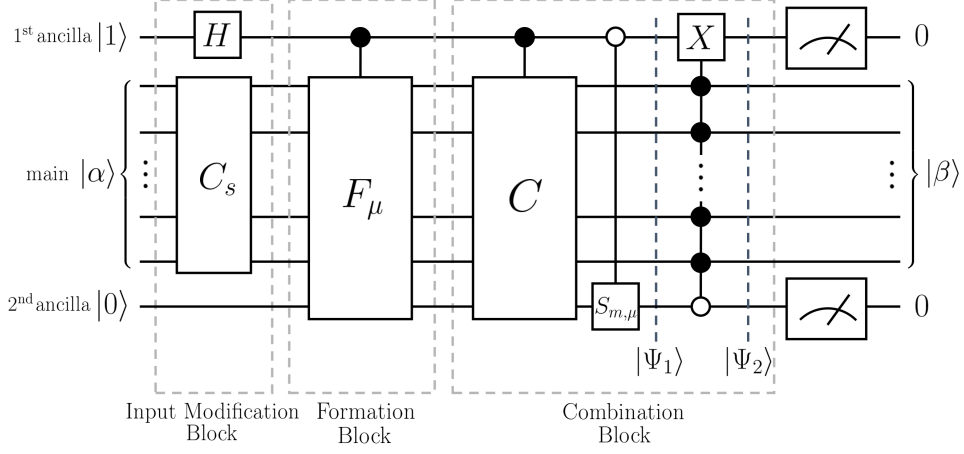


FIG. 2. Polynomial Representative Circuit (PRC) which is used to represent an operation of a companion matrix.

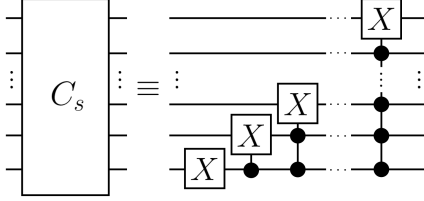


FIG. 3. The cyclic-swap gate can be implemented by a sequence of Toffoli gates.

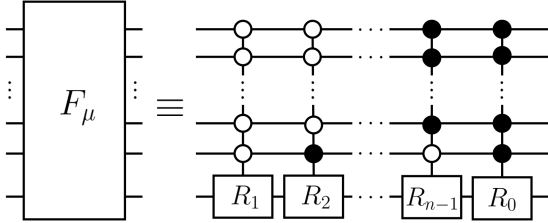


FIG. 4. The formation block can be simulated by a sequence of controlled-rotation gates.

This gate is, in fact, a rotation gate $R_y(\theta)$, where $\theta = 2\cos^{-1}(1/\sqrt{2^m}\mu)$. Scaling gate $S_{m,\mu}$ is to be operated on the second ancillary qubit in case that first ancillary qubit is in state $|0\rangle$. After the controlled- $S_{m,\mu}$ gate, the state is transformed into the following,

$$\begin{aligned} |\Psi_1\rangle &= \frac{1}{\sqrt{2}} \left[|0\rangle \otimes ((I^{\otimes m} \otimes S_{m,\mu})(C_s \otimes I)(|\alpha\rangle \otimes |0\rangle)) \right. \\ &\quad \left. - |1\rangle \otimes CF_\mu(C_s \otimes I)(|\alpha\rangle \otimes |0\rangle) \right] \\ &= \frac{1}{\sqrt{2^{m+1}\mu}} \left[|0\rangle \otimes [\alpha_1 \bullet \alpha_2 \bullet \dots \alpha_{n-1} \bullet \alpha_0 \bullet] \right] \end{aligned}$$

$$+ |1\rangle \otimes [\bullet \bullet \dots \bullet -\vec{a}' \cdot \vec{\alpha} \bullet]. \quad (22)$$

However, referring to (22), the final state is not exactly $|\beta\rangle$. The last task is just to swap between the coefficient α_0 and $-\vec{a}' \cdot \vec{\alpha}$ in (22) using the Toffoli gate in Fig. 2, which results in

$$|\Psi_2\rangle = \frac{1}{\sqrt{2^{m+1}\mu}} |0\rangle \otimes |\beta\rangle \otimes |0\rangle + |\psi^\perp\rangle \quad (23)$$

where $|\psi^\perp\rangle$ refers to the case that the ancillary qubits do not all give the result '0'. Finally, the post-selection only the results of both ancillary qubits gives '0', the output of the algorithm becomes $|\beta\rangle$ with the success probability of $\frac{1}{2^{m+1}\mu^2}$.

C. Polynomial Root-finding by Eigenvalue Estimation Technique

In order to find roots of the polynomial, we will use the circuit shown in Fig. 5. A controlled operation of the PRC by the phase qubit is denoted by $c-C_p$ in the figure. To describe the operation, we will firstly assume that the main qubits are initially prepared in an eigenstate $|\psi_j\rangle$ of the companion matrix. An expected state from the operation will be in the form

$$|\beta\rangle = |\lambda_j| e^{2\pi i \omega_j} |\psi_j\rangle. \quad (24)$$

Here we will assume that ω_j has a binary expansion in the form $\omega_j = 0.x_1x_2x_3\dots x_b$, where b is bit-precision. As in IPEA, the $c-C_p$ will be operated 2^{b-k} times in the k^{th} iteration as illustrated in Fig. 6. The result of the first iteration is given by

$$\frac{1}{2} \left(\frac{1}{\sqrt{2^m}\mu} \right)^{2^{b-1}} \left(|0\rangle|0\rangle|\psi_j\rangle|0\rangle + |\lambda_j|^{2^{b-1}} e^{2\pi i(0.x_b)} |1\rangle|0\rangle|\psi_j\rangle|0\rangle \right). \quad (25)$$

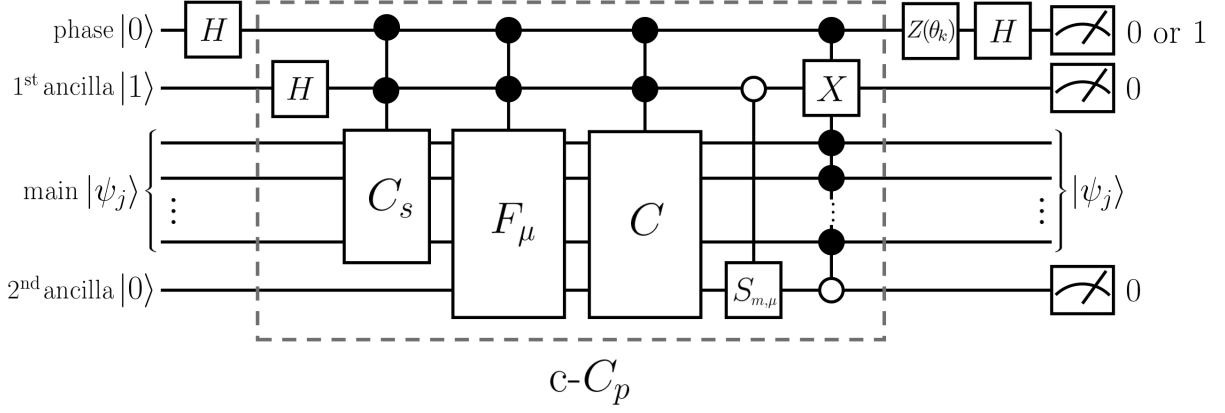


FIG. 5. A circuit scheme for finding polynomial roots.

Similarly, for the k^{th} iteration, we have

$$\frac{1}{2} \left(\frac{1}{\sqrt{2^m \mu}} \right)^{2^{b-k}} \left(|0\rangle|0\rangle|\psi_j\rangle|0\rangle + |\lambda_j|^{2^{b-k}} e^{2\pi i(0.x_{b-k+1}x_{b-k+2}\dots x_b)} |1\rangle|0\rangle|\psi_j\rangle|0\rangle \right). \quad (26)$$

After the operation of $Z(\theta_k)$ with $\theta_1 = 0$ and $\theta_k = 2\pi(0.0x_{b-k+2}x_{b-k+3}\dots x_b)$ followed by the Hadamard gate, the phase qubit will be in the following state,

$$\frac{1}{2\sqrt{2}} \left(\frac{1}{\sqrt{2^m \mu}} \right)^{2^{b-k}} \left((1 + |\lambda_j|^{2^{b-k}} e^{2\pi i(0.x_{b-k+1})}) |0\rangle + (1 - |\lambda_j|^{2^{b-k}} e^{2\pi i(0.x_{b-k+1})}) |1\rangle \right). \quad (27)$$

The value of x_{b-k+1} can be either 0 or 1. Therefore, the probabilities of finding the phase qubit in states $|0\rangle$ or $|1\rangle$ given that the both ancillary qubits give result 0 depend on the value of x_{b-k+1} ; namely,

$$P_0 = \frac{1 + 2 \cos(2\pi 0.x_{b-k+1}) |\lambda_j|^{2^{b-k}} + |\lambda_j|^{2^{b-k+1}}}{8 (2^m \mu^2)^{2^{b-k}}}, \quad (28)$$

$$P_1 = \frac{1 - 2 \cos(2\pi 0.x_{b-k+1}) |\lambda_j|^{2^{b-k}} + |\lambda_j|^{2^{b-k+1}}}{8 (2^m \mu^2)^{2^{b-k}}}. \quad (29)$$

Since x_{b-k+1} can be either 0 or 1, the value of $\cos(2\pi 0.x_{b-k+1})$ is either +1 or -1. In practice, x_b can be obtained by comparing P_0 and P_1 ²⁵, i.e., $x_b = 0$ if and only if $P_0 > P_1$; and $x_b = 1$ if and only if $P_0 < P_1$. In addition, the value of $|\lambda_j|$ from the k^{th} iteration can be calculated from the equation

$$|\lambda_j|^{2^{b-k+1}} = \left(4 (2^m \mu^2)^{2^{b-k}} |P_0 + P_1| - 1 \right). \quad (30)$$

It should be emphasized that in later iterations, the parameter θ_k used in $Z(\theta_k)$ is constructed from x_{b-k+1} from the prior iterations as in IPEA. Finally, an estimate of λ_j can now be obtained and the corresponding root of

the polynomial can be calculated depending on which mode (x -mode or $1/x$ -mode) is being used.

However, in general, the eigenstates of the companion matrix are unknown. The following approach is to estimate the greatest eigenvalue $|\lambda_{\max}|$. Suppose that an initial state is prepared in a mixed state, the density operator can be expressed as

$$\rho = \sum_j A_j |\psi_j\rangle\langle\psi_j|, \quad (31)$$

where A_j is a probability of preparing the initial state in the eigenstate $|\psi_j\rangle$. The operation of $c-C_p$ when the phase qubit is in state $|1\rangle$ transforms the density operator as

$$\rho \mapsto (C_p) \rho (C_p^\dagger). \quad (32)$$

For the k^{th} iteration after which $c-C_p^{2^{b-k}}$ is operated and followed by $Z(\theta_k)$ and the Hadamard gate, the probabilities of finding phase qubit in states $|0\rangle$ or $|1\rangle$ given that both ancillary qubits give result 0 are

$$P_0 = \sum_j A_j \left(1 \pm 2|\lambda_j|^{2^{b-k}} + |\lambda_j|^{2^{b-k+1}} \right) / \kappa, \quad (33)$$

$$P_1 = \sum_j A_j \left(1 \mp 2|\lambda_j|^{2^{b-k}} + |\lambda_j|^{2^{b-k+1}} \right) / \kappa. \quad (34)$$

where $\kappa = 8 (2^m \mu^2)^{2^{b-k}}$. In Equations (33) and (34), the terms with the largest eigenvalue $|\lambda_{\max}|$ will dominate if the number of iterations is large enough. Hence, the probabilities P_0 and P_1 will be reduced to the following forms:

$$P_0 \approx \left[1 + A_{\max} \left(|\lambda_{\max}|^{2^{b-k+1}} \pm 2|\lambda_{\max}|^{2^{b-k}} \right) \right] / \kappa, \quad (35)$$

$$P_1 \approx \left[1 + A_{\max} \left(|\lambda_{\max}|^{2^{b-k+1}} \mp 2|\lambda_{\max}|^{2^{b-k}} \right) \right] / \kappa. \quad (36)$$

The value of x_{b-k+1} can be found by comparing P_0 and P_1 . Even without knowing the probability A_{\max} , an

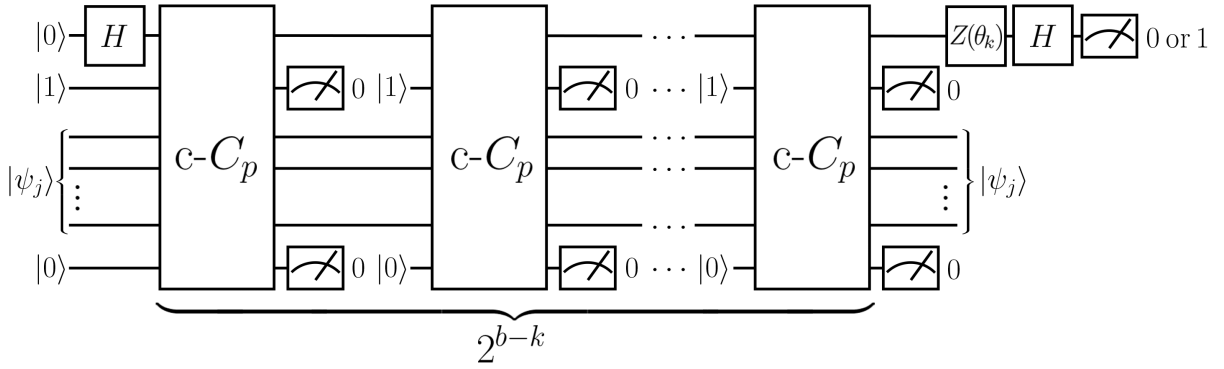


FIG. 6. In order to estimate polynomial roots up to b bit-precision, $c-C_p$ must be operated 2^{b-k} times in the k^{th} iteration.

estimate of $|\lambda_{\max}|$ can be obtained from

$$|\lambda_{\max}|^{2^{b-k}} = 2 \left(\frac{P_0 + P_1 - 2/\kappa}{|P_0 - P_1|} \right). \quad (37)$$

After the largest root is found, it can be factorized from the polynomial, and the same technique and procedure can be repeated to calculate the other roots.

III. DISCUSSIONS

In order to justify the efficiency of the quantum algorithm for finding roots of a polynomial, we have to compare it with the classical algorithm for solving the same problem. One of the most efficient classical algorithms for finding roots of a polynomial is created by Pan in 2002²⁸. We shall compare these two versions of the algorithm based on (i) resources required for calculation and (ii) algorithmic complexity.

We start by comparing the number of bits and qubits required by the computations. In order to find roots of an n^{th} degree polynomial, the quantum algorithm requires $O(\log n)$ qubits. In contrast, Pan's classical root-finding algorithm requires $O(n)$ or $O(n \log n)$ bits. This makes it obvious that, for large n , the quantum algorithm requires many fewer bits than its classical counterpart. In this way, the quantum version of algorithm is capable for finding roots of a much higher order degree than the classical one.

Next, we will compare their algorithmic complexities. In Pan's algorithm, the number of operations required to find the roots is

$$O((n \log^2 n)(\log^2 n + \log b)) \quad (38)$$

where b is the bit precision of the solutions. In contrast, since any m -qubit unitary gates can be simulated using only single-qubit gates and CNOT gates²⁹, it is appropriate to compute the complexity of a quantum circuit in terms of the number of single-qubit gates and CNOT gates required to construct the circuit. From polynomial

root-finding circuit, many unitary operations are controlled by several qubits. We will use the following corollary to calculate complexity of these gates (See Corollary 7.12 in Barenco *et al.*²⁴).

Corollary: *For any unitary U , the corresponding $c-U$ gate controlled by $(m-2)$ -qubit can be simulated by $O(m)$ basic operations in m -qubit network, where the initial value of one qubit is fixed and incurs no net change.*

In order to apply this corollary with the root-finding circuit, we need one more ancillary qubit and set its initial value to state $|0\rangle$. Note that this ancillary qubit can also be reused to simulate several controlled-unitary operations. In order to compute the overall complexity of the whole circuit, a complexity of each part may be found separately. The cyclic-swap gate can be simulated by m CNOT gates. Number of control qubits of each gate varies from 0 to $m-1$. For other CNOT gates with multiple control qubits, its operation can be simulated by $O(i+2)$ operations where $i > 1$ is the number of control qubits. Therefore the overall complexity of the cyclic-swap gate is $2 + \sum_{i=2}^{m-1} i \sim O(m^2)$. The formation gate consists of 2^m controlled rotation gates. Each gate, which is further controlled by m qubits, can be simulated by $O(m+2)$ basic operations. Hence the total complexity of the formation gate is $O(2^m m)$. The combination gate consists of m Hadamard gates and m NOT gates which brings its overall complexity to $O(m)$. Finally, the complexity of the $(m+1)$ -qubit CNOT gate before the application of the last Hadamard gate can be readily computed by the corollary above, which amounts to $O(m)$.

The total complexity of the complete circuit is the sum of complexity of each part as described above. However, in the complexity calculation, only the greatest term is kept. In the case of large n (where n is the degree of a polynomial), we can clearly see that the dominant term comes from the formation gate. Therefore, the complexity of the circuit in terms of the degree of a polynomial is

$$O(2^m m) \sim O(n \log n), \quad (39)$$

for one iteration, or

$$O(kn \log n), \quad (40)$$

for k iterations. Compared with the classical case for b -bit precision, our approach needs $k = 2^b$ and the total complexity of the quantum version is

$$O(2^b n \log n). \quad (41)$$

Here, we can validate that the quantum version of the algorithm is less complex than the classical version in the case where the polynomial has a high degree and a small number bit precision is required. On the contrary, the quantum algorithm may be an overkill when finding the roots, with high-bit precision, of a low degree polynomial.

IV. CONCLUSION

In summary, we have provided a quantum algorithm for finding roots of the n^{th} degree polynomial partially based on Daskin *et al.*'s circuit for finding complex eigenvalues of a general matrix¹⁸. To make a comparison with classical version of the algorithm, resources and algorithmic complexities are considered. The quantum version requires fewer number of (quantum) bits than its classical counterpart for a high degree polynomial. In terms of algorithm complexities, the quantum algorithm also trumps the classical algorithms for a high degree polynomial, requiring low bit-precision solutions. The growth

in complexity stems from a larger number of iterations needed to achieve the desired precision. Although our result clearly shows that finding the roots of polynomials using quantum information scheme is possible, the most important challenge, however, remains in strengthening the algorithm to overcome the classical algorithm both in the utilized resources and the chosen precisions. Another challenge lies of course in the practical issue of a working quantum computer. It is well known that the current quantum computer technology still falls short of the theoretical requirement of the algorithm, especially in terms of the number of entangled qubits and multiple-qubit quantum operations.

ACKNOWLEDGEMENT

We would like to thank Assist. Prof. Dr. Kwan Arayathanitkul for helpful discussions. This work is a collaboration of Collaborative Research Unit on Quantum Information, Mahidol University and Optical and Quantum Communication (OQC) Laboratory, National Electronics and Computer Technology Center (NECTEC). Grant No. 035/2557 from the Development and Promotion of Science and Technology Talents project (DPST) scholarship, research fund for DPST graduate with first placement is acknowledged.

-
- * Corresponding author's email: pruet.k@psu.ac.th
- ¹ E W Weisstein. *CRC encyclopedia of mathematics (2nd ed.)*. CRC Press Taylor & Francis, Boca Raton, Florida, 2009.
 - ² N Jacobson. *Basic algebra 1 (2nd ed.)*. Dover, 2009.
 - ³ J M McNamee. *Numerical Methods for Roots of Polynomials, Part 1*. Elsevier, Amsterdam, The Netherlands, 2007.
 - ⁴ Wankere R Mekwi. Iterative methods for roots of polynomials. Master's thesis, University of Oxford, 2001.
 - ⁵ Victor Y. Pan. Solving a polynomial equation: Some history and recent progress. *SIAM Review*, 39(2):187–220, 1997.
 - ⁶ P W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *Proc. 35th Annu. Symp. Foundations of Computer Science and IEEE Computer Society and Los Alamitos and CA*, pages 124–134, 1994. Ed. S. Goldwasser.
 - ⁷ Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.
 - ⁸ R. Feynman. Simulating physics with computers. *Int. J. Thy. Phys.*, 21:467, 1982.
 - ⁹ A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309:1704, 2005.
 - ¹⁰ X.-S. Ma, B. Dakic, W. Naylor, A. Zeilinger, and P. Walther. Quantum simulation of the wavefunction to probe frustrated heisenberg spin systems. *Nature Phys.*, 7:399–405, 2011.
 - ¹¹ Kenneth R. Brown, Robert J. Clark, and Isaac L. Chuang. Limitations of quantum simulation examined by simulating a pairing hamiltonian using nuclear magnetic resonance. *Phys. Rev. Lett.*, 97:050504, Aug 2006.
 - ¹² C. Negrevergne, R. Somma, G. Ortiz, E. Knill, and R. Laflamme. Liquid-state nmr simulations of quantum many-body problems. *Phys. Rev. A*, 71:032344, Mar 2005.
 - ¹³ Jonathan C. F. Matthews, Konstantinos Poullos, Jasmin D. A. Meinecke, Alberto Politi, Alberto Peruzzo, Nur Ismail, Kerstin Wörhoff, Mark G. Thompson, and Jeremy L. O'Brien. Observing fermionic statistics with photons in arbitrary processes. *Sci. Rep.*, 3, 03 2013.
 - ¹⁴ I Patu Ovidiu. Correlation functions and momentum distribution of one-dimensional hard-core anyons in optical lattices. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(1):P01004, 2015.
 - ¹⁵ Andrea Crespi. Suppression laws for multiparticle interference in sylvester interferometers. *Phys. Rev. A*, 91:013811, Jan 2015.
 - ¹⁶ A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem. *Coll. Comput. Complex.*, 3:3, 1996.
 - ¹⁷ M. Kacprowicz, R. Demkowicz-Dobrzanski, W. Wasilewski, K. Banzek, and I. A. Walmsley. Experimental quantum-enhanced estimation of a lossy

- phase shift. *Nature Photonics*, 4:357–360, 2010.
- ¹⁸ Anmer Daskin, Ananth Grama, and Sabre Kais. A universal quantum circuit scheme for finding complex eigenvalues. *Quantum Information Processing*, 13(2):333–353, 2014.
 - ¹⁹ Anmer Daskin, Ananth Grama, Giorgos Kollias, and Sabre Kais. Universal programmable quantum circuit schemes to emulate an operator. *The Journal of Chemical Physics*, 137(23):–, 2012.
 - ²⁰ Daniel S. Abrams and Seth Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.*, 83:5162–5165, Dec 1999.
 - ²¹ X-Q. Zhou, P. Kalasuwan, T. C. Ralph, and J. L. O’Brien. Calculating unknown eigenvalues with a quantum algorithm. *Nature Photonics*, 7:223–228, 2013.
 - ²² Hefeng Wang, Lian-Ao Wu, Yu-xi Liu, and F. Nori. Measurement-based quantum phase estimation algorithm for finding eigenvalues of non-unitary matrices. *Phys. Rev. A*, 82:062303, Dec 2010.
 - ²³ Stefanie Barz, Ivan Kassal, Martin Ringbauer, Yannick Ole Lipp, Borivoje Dakic, Alan Aspuru-Guzik, and Philip Walther. A two-qubit photonic quantum processor and its application to solving systems of linear equations. *Scientific Reports*, 4(5):6115EP, 2014.
 - ²⁴ A Barenco, C H Bennett, R Cleve, D P Di Vincenzo, N Margolus, P Shor, T Sleator, J A Smolin, and H Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457, 1995.
 - ²⁵ Robert B. Griffiths and Chi-Sheng Niu. Semiclassical fourier transform for quantum computation. *Phys. Rev. Lett.*, 76(17):3228–3231, Apr 1996.
 - ²⁶ J. Chiaverini, J. Britton, D. Leibfried, E. Knill, M. D. Barrett, R. B. Blakestad, W. M. Itano, J. D. Jost, C. Langer, R. Ozeri, T. Schaetz, and D. J. Wineland. Implementation of the semiclassical quantum fourier transform in a scalable system. *Science*, 308(5724):997–1000, 2005.
 - ²⁷ B. Buchberger. *An algorithmic method in polynomial ideal theory*. D. Reidel Publishing Company, Dordrecht Boston Lancaster, 1985.
 - ²⁸ Victor Y. Pan. Univariate polynomials: Nearly optimal algorithms for numerical factorization and root-finding. *Journal of Symbolic Computation*, 33(5):701 – 733, 2002.
 - ²⁹ Adriano Barenco. A universal two-bit gate for quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 449(1937):679–683, 1995.